

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ, МОЛОДЕЖИ И СПОРТА УКРАИНЫ

ХАРЬКОВСКАЯ НАЦИОНАЛЬНАЯ АКАДЕМИЯ

ГОРОДСКОГО ХОЗЯЙСТВА

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к самостоятельной работе

по дисциплине

«ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ОХРАНЕ ОКРУЖАЮЩЕЙ СРЕДЫ»

**Модуль 2. «Программные средства для обработки
экологической информации. Средства предоставления
и получения информации»**

(для студентов 2 курса (4 семестр) дневной формы обучения
направления подготовки 6.040106 «Экология, охрана окружающей среды
и рациональное природопользование»)

Харьков

ХНАГХ

2011

Методические указания к самостоятельной работе по дисциплине «Информационные технологии в охране окружающей среды». Модуль 2. «Программные средства для обработки экологической информации. Средства предоставления и получения информации» (для студентов 2 курса (4 семестр) дневной формы обучения направления подготовки 6.040106 «Экология, охрана окружающей среды и рациональное природопользование») / Харьк. нац. акад. гор. хоз-ва; сост.: Е. Г. Пономаренко, О. С. Ломакина. – Х.: ХНАГХ, 2011. – 58 с.

Составители: **Е. Г. Пономаренко,**
О. С. Ломакина

Рецензент: к. ф-м. н., доцент кафедры инженерной экологии городов
В. А. Баранник

Рекомендовано кафедрой инженерной экологии городов,
протокол № 2 от 25 сентября 2009 р.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	5
СОДЕРЖАТЕЛЬНЫЙ МОДУЛЬ 2.1. «ИСПОЛЬЗОВАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ»	6
1. БАЗЫ ДАННЫХ: ОБЩИЕ СВЕДЕНИЯ И СТРУКТУРА	6
1.1. Общие сведения	6
1.2. Реляционные базы данных	6
1.3. СУБД Microsoft Access	8
2. ОСНОВНЫЕ ЭТАПЫ ПРОЕКТИРОВАНИЯ БАЗ ДАННЫХ	10
3. СОЗДАНИЕ ТАБЛИЦ БАЗ ДАННЫХ	12
3.1. Создание таблиц в режиме конструктора	12
3.2. Различия между базами данных Access и таблицами Excel	18
4. СВЯЗЫВАНИЕ ТАБЛИЦ	20
4.1. Причины связывания таблиц	20
4.2. Правила связывания таблиц	21
4.3. Механизм создания связей между таблицами	21
4.4. Типы связей	22
4.5. Обеспечение целостности данных	24
4.6. Внесение изменений в схему связей	25
5. МАСКИ ВВОДА ДАННЫХ	27
6. СОЗДАНИЕ ЗАПРОСОВ	29
6.1. Общие сведения	29
6.2. Виды запросов	30
6.3. Создание запроса – выборки	32
6.4. Условия отбора записей	34
6.5. Построитель выражений	36
6.6. Вычисляемые поля	39
6.7. Запросы с параметром	40
6.8. Многотабличные запросы	40
6.9. Просмотр результата запроса	41
7. ФОРМЫ	42
7.1. Назначение и виды форм	42

7.2. Редактирование формы	43
8. ОТЧЕТЫ	45
8.1. Назначение и создание отчетов	45
8.2. Разделы отчета	45
СОДЕРЖАТЕЛЬНЫЙ МОДУЛЬ 2.2. «СОЗДАНИЕ ПРЕЗЕНТАЦИЙ В MS POWER POINT»	46
9. НАЗНАЧЕНИЕ И СТРУКТУРА ПРЕЗЕНТАЦИИ MS POWER POINT	46
9.1. Общие сведения	46
9.2. Структура окна Power Point	46
10. ПОСЛЕДОВАТЕЛЬНОСТЬ РАБОТЫ НАД ПРЕЗЕНТАЦИЕЙ	48
11. РЕЖИМЫ РАБОТЫ В POWER POINT	49
12. СОЗДАНИЕ ПРЕЗЕНТАЦИИ	51
12.1. Создание презентации в режиме структуры	51
12.2. Создание отдельного слайда	52
12.3. Размещение объектов на слайде	53
12.4. Работа с текстом	53
12.5. Анимация объектов слайда	53
12.6. Построение последовательности слайдов	54
12.7. Шаблон презентации	55
12.8. Мастер-слайд	57
12.9. Установка режимов демонстрации	57

ВВЕДЕНИЕ

В работе современных инженеров и исследователей часто возникает необходимость хранения и обработки большого количества информации. Также часто возникает необходимость в предоставлении результатов своей работы широкой аудитории. Поэтому основными заданиями, которые ставятся перед студентом в рамках изучения курса «Информационные технологии в охране окружающей среды» (модуль «Программные средства для обработки экологической информации. Средства представления и получения информации»), являются формирование современного уровня информационной и компьютерной культуры и овладение методами и приемами использования современных компьютерных средств и технологий для решения профессиональных задач.

В связи с тем, что учебным планом не предусмотрены лекционные занятия, целью самостоятельной работы в рамках курса «Информационные технологии в охране окружающей среды» является изучение терминологического аппарата и основ работы с программными продуктами *СУБД Microsoft Access* и *MS Power Point*, которые позволяют решить задания, названные выше.

Это обуславливает содержание и структуру данных методических указаний. Поэтому последние содержат теоретические основы работы с *СУБД Microsoft Access* и *MS Power Point*. Для контроля усвоения материала в конце каждого раздела приведены вопросы для самопроверки или задания для самоконтроля.

Полученные теоретические знания закрепляются в лабораторных работах при выполнении конкретных заданий.

СОДЕРЖАТЕЛЬНЫЙ МОДУЛЬ 2.1.

ИСПОЛЬЗОВАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ

1. БАЗЫ ДАННЫХ: ОБЩИЕ СВЕДЕНИЯ И СТРУКТУРА

1.1. Общие сведения

База данных (БД) - это один или несколько файлов, закодированных специальным образом. Компьютерная программа, которая управляет базой данных, называется *системой управления базой данных*, или СУБД.

По сути, БД - это некоторое подобие электронной картотеки, электронного хранилища данных, которое хранится в компьютере в виде одного или нескольких файлов. При этом возникает необходимость в выполнении ряда операций с БД. В первую очередь это:

- добавление новой информации в существующие файлы БД;
- изменение (модификация) информации в существующих файлах БД;
- поиск информации в БД;
- удаление информации из существующих файлов БД.

1.2. Реляционные базы данных

Концепция баз данных менялась на протяжении длительного времени. Наиболее распространенной стала реляционная модель. *Реляционная база данных* - это база данных, в которой все данные, доступные пользователю, организованы в виде взаимосвязанных таблиц, а все операции над данными сводятся к операциям над этими таблицами.

У каждой таблицы должно быть уникальное имя.

Таблицы разделены на строки и столбцы, на пересечении которых содержатся значения данных. Строки в таблицах обычно называют *записями*, а столбцы - *полями*.

Чтобы отличать одну строку от другой, используют понятие *ключа*. Ключом называется столбец или группа столбцов таблицы, значения которых *уникальным образом* идентифицируют каждую строку таблицы. У таблицы может

быть только один ключ. Все записи в ключевом поле (и существующие и *возможные в будущем*) должны быть разными, не важно какими – *но разными*. Именно по ключевому полю различают записи в таблице.

Индекс - это средство, обеспечивающее быстрый доступ к строкам таблицы на основе значений одного или нескольких столбцов. По умолчанию, когда пользователь просматривает данные, которые сохраняются в таблице, они отображаются в том порядке, в котором лежат в базе - все сведения из одной записи появляются в одной строке, но записи никак не отсортированы. Это значит, что для поиска интересующей записи придется «прокрутить» весь список. К счастью, реляционные базы данных предоставляют средство для быстрого поиска нужной информации, а именно индексы. СУБД пользуется индексом так же, как человек пользуется предметным указателем книги. В индексе хранятся значения данных и указатели на строки, где эти данные встречаются. Значения данных в индексе располагаются в убывающем или возрастающем порядке, чтобы СУБД могла быстро найти требуемое значение.

В сущности, индексы таблиц - это отсортированные списки. Каждый элемент индекса указывает на некоторую запись в таблице. Поэтому при поиске записи система управления базами данных сначала просматривает индекс, что занимает совсем немного времени. Индекс можно построить по полю почти любого типа. Индексы значительно убабстряют поиск данных.

Структура реляционной БД задается *схемой данных*. Схема данных включает в себя определение структуры каждой таблицы и определение связей между таблицами. Схема данных в Access создается специально для использования связей между таблицами при обработке данных из разных таблиц, а также для обеспечения целостности БД.

Для работы с БД используют специальные языки, в целом называемые языками баз данных. В реляционных СУБД поддерживается единый язык, содержащий все необходимые средства - от создания БД до обеспечения пользовательского интерфейса при работе с данными - язык SQL (Structured Query Language).

1.3. СУБД Microsoft Access

С одной стороны, под Access мы будем понимать непосредственно базу данных (файл с расширением MDB, сохраненный на диске). С другой стороны, под Access будем понимать систему управления реляционной БД, которая включает в себя все необходимые инструментальные средства для создания БД.

Access ориентирован на работу с объектами, к которым относятся таблицы БД, запросы, а также объекты приложений для работы с БД: формы, отчеты, страницы, макросы и модули.

Все объекты представлены в окне базы данных Access. Все операции по работе с объектами БД будут начинаться с данного окна. Ниже приводиться краткое описание объектов Access.

Таблицы - создаются пользователем для *хранения* данных. Эти данные могут изменяться только при корректировке записей баз данных, но в ячейках базовых таблиц принципиально не может быть вычисляемых значений. Таблица состоит из полей и записей. Все записи состоят из одинаковых полей, а характер информации для одного поля во всех записях одинаковый, но разные поля могут иметь разный тип и/или атрибуты хранимых данных.

Запросы - создаются пользователем для выборки нужных данных из одной или нескольких связанных таблиц. Результатом выполнения запроса является таблица, которая может быть использована наряду с другими таблицами БД при обработке данных. С помощью запросов можно также управлять строением БД, к примеру - создавать новые таблицы на основе уже существующих. То есть запросы — это те же самые таблицы, только они заполняются не вручную, а с помощью заранее заданных зависимостей. Таким образом, то, что нельзя в таблице, можно в запросе, например, выполнять математические и логические операции. Свое название запросы получили оттого, что они схожи с широко распространенным естественным действием аналогичного назначения.

Формы - основное средство для создания диалогового интерфейса приложения с пользователем. Форма, к примеру, может создаваться для ввода и просмотра взаимосвязанных данных на экране в удобном виде. Обычно в рабо-

чих БД пользователь не имеет прямого доступа к таблицам, а только через формы. Access связывает каждую форму с ее источником (таблицами, запросами), что обеспечивает соответствие данных в форме и источнике.

Отчеты - предназначены для формирования выходных документов, содержащих, к примеру, результаты решения задач пользователя для дальнейшего вывода их на печать. Позволяют также осуществлять некоторые простые расчеты.

Страницы - Web-страницы, поддерживающие связь с базой данных и позволяющие просматривать, редактировать и вводить данные в БД, работая в окне программы-браузера.

Макросы - программы, которые содержат описание последовательности действий, выполняемых при наступлении некоторого события в объекте или элементе управления приложения. Каждое действие реализуется макрокомандой.

Модули - содержат процедуры, записанные на языке VBA (Visual Basic for Application).

Вопросы для самопроверки:

1. Что собой представляет база данных?
2. Перечислите основные операции, которые можно выполнять с базами данных
3. Для чего в таблицах БД используется ключ?
4. Для чего в БД используется индекс?
5. Для чего используется схема данных?
6. Могут ли таблицы (и в каких случаях):
 - а) не содержать ключевого поля; б) не содержать индексированного поля; в) содержать несколько индексированных полей?
8. Какой элемент БД используется для внесения и изменения информации?
9. Какой элемент базы данных служит для хранения информации?
10. Какой элемент БД может содержать вычисляемые поля?
11. В чем заключается отличие таблицы от запроса?

2. ОСНОВНЫЕ ЭТАПЫ ПРОЕКТИРОВАНИЯ БАЗ ДАННЫХ

Этап 1. Определение цели создания базы данных. На этом этапе определяются основные функции БД, темы таблиц, отчетов, форм. Производится сравнение с аналогами.

Этап 2. Определение списка таблиц. Это самый сложный и ответственный этап. От того насколько продуман список таблиц, зависит эффективность работы БД. Формально для работы базы данных достаточно только одной таблицы. Однако работа с одной большой таблицей неудобна. Такие таблицы громоздки и перегружены информацией. Реляционные базы данных позволяют разбить всю информацию на серию таблиц с относительно небольшим количеством полей и связать эти таблицы между собой. Процесс создания структуры таблиц, обеспечивающих наиболее эффективную работу БД, называется *нормализацией данных*.

Этап 3. Определение необходимых в таблице полей. В нормализованной БД каждая таблица содержит информацию на отдельную тему, а каждое поле в таблице содержит отдельные сведения по теме таблицы. Каждое поле должно быть связано с темой таблицы. В таблице должна присутствовать вся необходимая информация. Информацию следует разбивать на наименьшие логические единицы (Например, поля «Фамилия», «Имя» и «Отчество», а не общее поле «ФИО»).

Этап 4. Задание ключевых полей. Для того чтобы Access мог связать данные из разных таблиц, каждая таблица должна содержать ключевое поле.

Этап 5. Определение связей между таблицами.

Этап 6. Обновление структуры базы данных. После проектирования таблиц, полей и связей необходимо еще раз просмотреть структуру базы данных и выявить возможные недочеты. Это важно, так как после заполнения таблиц преобразование структуры будет делать гораздо сложнее. Проверка делается путем тестирования базы на маленьком объеме информации.

Этап 7. Добавление данных и создание других объектов базы данных. На этом этапе создаются все запросы, формы, отчеты, макросы и модули

Этап 8. Использование средств анализа в Microsoft Access. В Access есть два инструмента для улучшения структуры баз данных. Они вызываются командой *Сервис\Анализ*. Мастер анализа таблиц исследует таблицу, в случае необходимости предлагает новую ее структуру и связи, а также переделывает ее. *Анализатор быстрого действия* исследует всю базу данных, дает рекомендации по ее улучшению, а также осуществляет их.

Задания для контроля:

Продумайте и изобразите схематически предварительную структуру базы данных (названия таблиц; поля, содержащиеся в таблицах и связи между таблицами), которая бы содержала следующую информацию: состав вашей группы, полные анкетные данные студентов, перечень изучаемых дисциплин, успеваемость студентов, наличие задолженностей по дисциплинам. Результат покажите преподавателю.

3. СОЗДАНИЕ ТАБЛИЦ БАЗ ДАННЫХ

Процесс разработки БД начинается с задания описания структур таблиц. Это своеобразный каркас, который создается до того, как таблица начнет наполняться данными.

3.1. Создание таблиц в режиме конструктора

Создание таблиц целесообразно в режиме *Конструктора*, а любое изменение структуры таблиц – *только в режиме Конструктора*.

	Имя поля	Тип данных	Описание
▶			

Свойства поля

Общие Подстановка

Окно конструктора состоит из пяти основных элементов. Сверху располагаются три столбца, озаглавленные *Имя поля*, *Тип данных*, *Описание*. В нижней части окна есть две вкладки: *Общие* и *Подстановка*.

Столбец *Описание* имеет лишь вспомогательный характер и служит для комментариев. Все, что в нем набирается, нужно только вам самим (например, чтобы через некоторое время не забыть, для чего служит то или иное поле).

В столбце *Имя поля* пользователь имеет возможность задать любое имя столбца, какое посчитает необходимым. Следует помнить, что столбцы в таблице будут располагаться именно в том порядке, в котором вы их задали. Таким образом, желательно с самого начала продумать их последовательность с учетом удобства последующей эксплуатации всей таблицы в целом. Так любая сортировка таблиц практически во всех СУБД производится слева направо. Иными словами, сначала выполняется условие сортировки, заданное в первом

столбце, затем - во втором, и так далее. Если в таблице, по недосмотру, столбец с номерами домов окажется перед столбцом с названиями улиц, то вы не сможете, без дополнительных вспомогательных операций, отсортировать таблицу сначала по улицам, а потом по номерам домов. Придется создавать отдельный запрос, который потребует дополнительного места, дополнительной вычислительной мощности и дополнительного времени на его выполнение.

После задания имени поля в СУБД Microsoft Access требуется непременно указать тип данных, которые в этом поле предполагается хранить. В Access допускается использование девяти типов данных:

Текстовый (тип данных по умолчанию) - текст или цифры, не участвующие в расчетах. Число символов в поле (включая пробелы) не должно превышать 255. Максимальное число символов, которое можно ввести в поле, задается в свойстве *Размер поля*. Пустые символы в неиспользуемой части поля не сохраняются.

Поле МЕМО - длинный текст, например некоторое описание или примечание. Максимальная длина 64000 символов. Для полей типа *МЕМО* не допускается сортировка и индексирование.

Числовой - числовые данные, используемые в математических вычислениях. Конкретные варианты числового типа и их длина задаются в свойстве *Размер поля*.

Денежный - для проведения денежных расчетов.

Дата/Время - Значения даты или времени, относящиеся к годам с 100 по 9999 включительно.

Счетчик - тип данных, который для каждой новой записи автоматически вводит уникальные целые, последовательно возрастающие (на 1), числа. Значения этого поля нельзя изменить или удалить. *В таблице не может быть более одного поля типа счетчик*. Обычно используется для задания ключевого поля таблицы.

Логический - логические данные, которые могут иметь одно из двух возможных значений *Да/Нет*; *Истина/Ложь*; *Вкл./Выкл.*

Поле объект OLE - это объект OLE (Object Linking and Embedding), связанный или внедренный в таблицу Access. Поля данного типа не хранят информацию как таковую, а содержат ссылки на объекты, которые могут быть включены в БД с использованием OLE-протокола обмена данными. Такими объектами могут быть, например, графические файлы. Даже самая лучшая СУБД не в состоянии обеспечить хранение всех существующих на сегодняшний день видов информации, что, впрочем, и не нужно. Достаточно осуществить связь с приложением, работающим с тем или иным типом файлов. Для этого и существует протокол OLE. При заполнении таблицы, для того чтобы вставить в ячейку поля OLE объект, надо установить на нее курсор и, нажав правую кнопку мыши, вызвать контекстное меню, в нем выбрать команду *Добавить объект*. Объект либо отождествляется с приложением, в котором он может быть создан или отредактирован либо вводится прямо из файла. Для полей типа OLE не допускается сортировка и индексирование.

Гиперссылка - в качестве гиперссылки можно указывать путь к файлу на жестком диске, адрес URL. Максимальная длина 64 000 символов.

Мастер подстановок - выбор этого типа данных запускает мастер подстановок. Мастер строит для поля список значений на основе поля из другой таблицы. Значения в такое поле будут вводиться из одного из полей списка. Проще говоря, данный тип позволяет определить вычисляемое поле.

Указав тип поля, вы попадаете в список возможных значений (атрибутов), которые приводятся на вкладке *Общие*. Для каждого типа данных содержание этой вкладки имеет свои отличия. Допустим, вы создаете поле, в котором предполагается вводить наименование городов, где живут студенты академии. Естественно, это будет поле текстового типа. Однако отводить ему все двести пятьдесят пять символов наверняка излишне. Даже если очень постараться, трудно будет найти такое наименование населенного пункта, общая длина которого превысит тридцать букв. Таким образом, указав на вкладке *Общие* в строке *Размер поля* величину 30, вы сэкономите место для 225 символов. В том случае, если данные могут иметь определенный формат, то их пред-

ставление можно принудительно указать в строке *Формат поля*. Допустим, для полей типа *Дата/Время* можно указывать только часы, либо часы и минуты, либо часы, минуты и секунды, причем как в 12-и, так и в 24-х часовом представлении. Все возможные варианты представления данных автоматически подбираются в зависимости от того, какой тип данных вами выбран. Строка *Маска ввода* существует на тот случай, когда данные должны вводиться в заранее заданной форме. Человек привык записывать телефонный номер, указывая в скобках междугородний код, затем несколько групп цифр, разделенных тире. Было бы удобно, если бы система не заставляла оператора вводить каждый раз, кроме цифр, еще и скобки и тире. Поэтому весьма нелишне ввести маску набора цифр, чтобы Microsoft Access автоматически расставляла вводимые с клавиатуры цифры согласно необходимой схеме. Существуют маски и для других типов данных (о масках будет подробнее рассказано дальше). То, что набрано в строке *Подпись*, Microsoft Access в дальнейшем использует как всплывающую подсказку для элемента управления. То есть, при наведении маркера мыши на тот или иной элемент возникает белый прямоугольник с краткой пояснительной надписью. Строка *Значение по умолчанию* позволяет автоматически заполнять поле некоторым значением, которое встречается часто. Так, в ХНАГХ много студентов из Харькова. Тогда стоит ввести в строке *Значение по умолчанию*, если придерживаться вышеописанного примера, слово "Харьков". В этом случае, как только будет создана новая запись, в поле "Город" по умолчанию окажется введенным Харьков. Естественно, его можно заменить любым другим, однако больше не придется постоянно следить за этим полем без особой надобности. Строка *Условие на значение* предназначена для организации так называемой "защиты от дурака". Желательно постараться заранее ограничить все возможные действия пользователя, которые могут привести к нежелательным или даже разрушительным последствиям (в тех случаях, когда это возможно). Например, нельзя заплатить отрицательную сумму. Или, наверняка, учащийся у нас студент не мог родиться до 1900 года (столько не живут). Заранее определив "что нельзя", вы одновременно убиваете двух зайцев. С одной

стороны, это воспрепятствует попаданию в таблицу заведомо неверных данных, даже если они допустимы по типу. С другой стороны, в случае возникновения подобной ошибки Microsoft Access выведет на экран соответствующее предупреждение. Это предупреждение можно сформулировать самому, заполнив атрибут *Сообщение об ошибке*. Как известно, любые данные могут быть важными, а могут быть второстепенными. Так, фамилия, имя, отчество студента – важные данные, а его семейное положение – второстепенные. Важные данные обязательно должны присутствовать в таблице. Чтобы система была в состоянии самостоятельно отслеживать заполнение важных полей, следует сказать *Да* в строке *Обязательное поле*. Тогда Microsoft Access не внесет в таблицу строку, если хотя-бы одно из обязательных полей оказалось не заполнено. В строке *Индексированное поле* задается *Да*, если поле будет использоваться в качестве индексного (об индексах смотри выше). Если индексное поле предполагается использовать в качестве ключевого, то должен быть выбран режим, не допускающий совпадений. Атрибуты *Сжатие Юникод*, *Режим ИМЕ* и *Режим предложений ИМЕ* чаще всего не изменяются (используются при работе с восточноазиатскими языками). Атрибут *Смарт-теги* представляет собой дополнительные инструменты которые встраиваются в поле и позволяют открыть какое-либо приложение Office (например, календарь встреч из Outlook).

В Microsoft Access существует правило, в соответствии с которым таблица должна содержать ключевое поле. Ключевым должно быть такое поле или их сочетание, которое однозначно описывает всю запись (то есть строку) в таблице (то есть не повторяется и не может повторяться). Если среди полей таблицы таких нет, то оно создается специально. Ключ не может содержать пустых значений (Null). Значение Null означает, что в поле нет никаких данных, например, потому, что они неизвестны. Значение Null нельзя приравнивать к строке, содержащей пробелы.

Данные ключевого поля используются для индексирования таблицы, что ускоряет поиск и обработку информации. При включении новых записей или

удалении старых записи таблицы не перемешаются, изменяется только местоположение каждого индекса.

Ключ используется для связывания одной таблицы с другой.

Чтобы поле стало ключевым, в режиме конструктора его следует выделить (т.е. выделить всю строку этого поля) и нажать мышью на изображение ключа на панели инструментов *Конструктор таблиц*. Слева, возле данного поля, появится маленький ключик, который и символизирует, что данное поле является ключевым. Если же Вы попытаетесь сохранить таблицу без задания ключевого поля, то Access предложит Вам задать такое поле или сделает это сам, дополнив таблицу полем типа *Счетчик* и назначив его ключевым.

Наименование типа данных	Содержимое	Значения данного типа
1	2	3
Текстовый (Тип данных, используемый по умолчанию)	Текст или числа, не требующие проведения расчетов, например, номера телефонов.	255 символов, но не более значения атрибута <i>Размер поля</i> .
Поле МЕМО	Длинный текст или комбинация текста и чисел.	До 65535 символов.
Числовой	Числовые данные, используемые для проведения расчетов.	Байт - целые числа от 0 до 255; Целое - целые числа от 32768 до 32767. Длинное целое - целые числа от 2147483648 до 2147483647; Одинарное с плавающей точкой - числа в пределах от $-3.402821 \cdot 10^{38}$ до $3.402821 \cdot 10^{38}$; Двойное с плавающей точкой числа в пределах от $-1.79769313486231 \cdot 10^{308}$ до $1.79769313486231 \cdot 10^{308}$

Продовжения табл.

1	2	3
Дата/время	Даты и время, относящиеся к годам с 100 по 9999 включительно.	Формат даты и времени определяется региональными настройками Windows. Можно изменить вручную с помощью атрибута <i>Формат поля</i> .
Денежный	Числовые данные снабженные обозначением денежной единицы, используемые в математических расчетах, проводящихся с точностью до 15 знаков в целой и до 4 знаков в дробной части.	Формат денежной единицы определяется региональными настройками Windows. Можно изменить вручную с помощью атрибута <i>Формат поля</i> . Наименование денежной единицы задается только настройками Windows.
Счетчик	Уникальные последовательно возрастающие (на 1) или случайные числа, автоматически вводящиеся при добавлении каждой новой записи в таблицу. Значения полей типа счетчика обновлять и изменять вручную нельзя.	4 байт
Логический	Логические значения, а также поля, которые могут содержать одно из двух возможных значений (True/False, Да/Нет).	1 бит.
Поле объекта OLE	Объект (например, электронная таблица Microsoft Excel, документ Microsoft Word, рисунок, звукозапись), связанный или внедренный в таблицу Microsoft Access.	Ограничивается объемом диска.
Гиперссылка	Строка, состоящая из букв и цифр и представляющая адрес файла или Интернет-ресурса.	
Мастер подстановок	Создает поле, в котором предлагается выбор значений из списка или из поля со списком, содержащего набор постоянных значений или значений из другой таблицы. Выбор этого параметра в списке в ячейке запускает мастера подстановок, который определяет тип поля.	Тот же размер, что и у ключевого поля, используемого в подстановке.

3.2. Различия между базами данных Access и таблицами Excel

На первый взгляд, таблицы Access очень похожи на таблицы Excel. Однако между ними существуют принципиальные различия:

- Для каждого столбца (поля) таблицы Access определен единый тип данных, тогда как в столбец таблицы Excel можно вводить данные разных типов;

- Access позволяет не просто вводить данные в таблицы, но и контролировать правильность этих данных. Для этого устанавливаются правила проверки на уровне таблиц. При этом, как бы вы ни вводили данные — прямо в таблицу, с помощью формы или другим способом - Access не позволит сохранить в записи данные, нарушающие эти правила;
- В таблицах Excel часто приходится держать повторяющиеся данные. Структуру таблиц базы данных разрабатывают таким образом, чтобы избежать дублирования информации. Это позволяет не только экономить память, но и увеличить скорость и точность обработки данных;
- Access имеет очень развитую систему защиты от несанкционированного доступа, позволяя разным пользователям или группам пользователей назначать разные права доступа, как к данным, так и к элементам интерфейса: формам, командам меню, отчетам;
- Excel обладает прекрасными возможностями выполнения расчетов и анализа данных непосредственно в таблицах. В Access же в таблицах данные только хранятся, а для их обработки (в том числе вычисления и анализа данных) используются запросы.

Вопросы для самопроверки:

2. В каком порядке производится сортировка столбцов в таблице?
3. Какие типы данных позволяют вносить текстовую информацию?
4. Какие типы данных позволяют вносить цифровую информацию?
5. Какие типы полей содержат: а) только уникальные данные; б) повторяющуюся информацию.
6. Для чего используется маска ввода?
7. Может ли индексное поле содержать повторяющиеся данные?
8. Поясните различие в использовании таблиц, если выставлены установки:
а) «Обязательное поле. Да»; б) «Обязательное поле. Нет».

Задания для контроля:

Самостоятельно задайте структуру таблицы, содержащей результаты контроля выбросов загрязняющих веществ в атмосферный воздух

№	Наименование загрязняющего вещества	ПДКс.с.	Концентрация вещества, мг/м ³	Дата контроля	Ф. И. О. проверяющего
1	Оксид азота	0,06	0,04	10.05.2010	Иванова А. А.
2	Сажа	0,05	0,055	12.5.2010	Петров В. В.
3	Аммиак	0,04	0,02	10.05.2010	Иванова А. А.

4. СВЯЗЫВАНИЕ ТАБЛИЦ

4.1. Причины связывания таблиц

Чаще всего причиной размещения данных в двух и более таблицах в одной БД является дублирование данных (наличие в таблице повторяющихся фрагментов). Для того чтобы сэкономить место на диске и в памяти, были разработаны принципы *нормализации*, которые позволили исключить дублирование. Большинство пользователей, работающих с базой данных на своем компьютере, не сталкивается с необходимостью экономии места на диске, и нормализация выполняется ради соблюдения другого принципа: данные должны быть организованы таким образом, чтобы в случае их редактирования каждое исправление или дополнение делалось только в одном месте базы данных. Третья важная причина разнесения информации по таблицам - необходимость группировки данных по их содержанию. Каждая таблица может отвечать определенной тематике. Четвертая причина - принцип модульности, который предполагает, что таблицы, подготовленные для одной базы данных, можно использовать и в других.

Таким образом, хранение данных в связанных таблицах обладает рядом преимуществ:

- экономия времени, поскольку одни и те же данные не приходится вводить в нескольких таблицах;
- уменьшение размера базы данных (порой весьма значительное по сравнению с размером плоской таблицы), которое экономит дисковое пространство на вашем компьютере и облегчает перенос базы данных;
- существенное сокращение количества ошибок. Представьте себе, что вам приходится вводить одни и те же данные в большом количестве записей. Сколько раз вы сможете набрать слово «Эвтрофирование» в поле *Тема занятий* без единой ошибки? Если же повторяющиеся данные хранятся в связанной таблице, то информацию достаточно ввести всего один раз; после этого в исходной таблице для повторяющихся данных вводится только код (обычно короткая

цифровая или алфавитно-цифровая последовательность). Вы даже можете настроить поле так, чтобы выбрать код из списка и обойтись вообще без набора текста!

- готовые таблицы базы данных можно перенести в другие базы данных и их не надо будет заполнять с «нуля».

Для того чтобы связать две таблицы, нужно в первую очередь предусмотреть поля в таблицах, значения которых пригодны для связи.

4.2. Правила связывания таблиц

Существуют следующие правила для установления связей между таблицами:

- таблицы должны содержать поля с одинаковыми записями; Но при этом связанные поля не обязательно должны иметь одинаковые названия;
- типы данных и длины полей, а главное — значения обязательно должны совпадать. Существует исключение. Поле типа *Счетчик* может быть связано с числовым полем, если для свойства *Длина поля* у него установлено значение *Длинное целое*

Необходимо также точно знать, могут ли быть в поле повторяющиеся значения. Если есть - значит «многие», если нет - значит «один» (подробно варианты связей разбираются ниже).

Связь между таблицами устанавливает отношения между совпадающими значениями в ключевых полях.

4.3. Механизм создания связей между таблицами

Механизм описания логических связей между таблицами в Access реализован в виде объекта, называемого *Схемой данных*. Перейти к созданию связей между таблицами можно (находясь на закладке *Таблицы*) через меню *Сервис\Схема данных* или с помощью кнопки *Схема данных* на панели инструментов. После этого надо включить таблицы в схему данных. Для этого используется команда *Добавить таблицу*. Установка связи производится перетаскива-

нием имени одного поля на имя другого. Указатель мыши в момент завершения перетаскивания должен принять вид прямоугольника.

Линии между окнами таблиц изображают связи между отдельными полями. Символы рядом с концами линии обозначают тип связи. Их смысл будет рассмотрен ниже. При работе с окном *Схема данных* важно расположить таблицы таким образом, чтобы линии связей были четко видны. Окна таблиц в окне *Схема данных* можно перетаскивать мышью за заголовок. Всегда располагайте окна таблиц так, чтобы линии не запутывались и были наглядными.

Когда в правой части окна таблицы появляется полоса прокрутки, это значит, что отображаются не все имена полей. Чтобы избавиться от полосы прокрутки и видеть все поля, можно увеличить высоту окна таблицы перетаскиванием его верхнего или нижнего края. Аналогично изменяется и ширина окна таблицы, что позволяет выводить полные имена полей.

4.4. Типы связей

Связь «один-ко-многим»

Связь "один ко многим" - наиболее распространенный вид связи. Создается в том случае, если в первой из таблиц поле имеет уникальное значение индекса (то есть Access требует, чтобы его значение никогда не повторялось в различных записях), а во второй таблице записи в этом поле могут повторяться. Уникальное поле является либо ключевым (чаще всего именно так и бывает), или же его свойство *Индексированное поле* имеет значение *Да (совпадения не допускаются)*. Поле второй таблицы в типичной связи не обладает уникальным значением индекса, то есть его свойство *Индексированное поле* имеет значение *Нет* или *Да (допускаются совпадения)*. Таблица, содержащая уникальное поле, обычно называется *базовой*, а вторая таблица - *подчиненной*. Поскольку одной записи базовой таблицы может соответствовать несколько записей в подчиненных таблицах, подобная связь называется "один-ко-многим". При такой связи каждой строке базовой таблицы может соответствовать множество строк под-

чиненной таблицы. Однако каждой строке подчиненной таблицы может соответствовать только одна строка базовой таблицы.

В Microsoft Access сторона связи "один-ко-многим", которая соответствует базовой таблице, обозначается символом **1**. Сторона связи, которой соответствует подчиненной таблице, обозначается символом бесконечности ∞ .

Связь «один-к-одному»

Эта связь подразумевает, что в обеих таблицах в связанных полях имеются только уникальные записи. На первый взгляд это кажется абсурдным: не проще ли было создать одну таблицу вместо двух? Действительно, в некоторых случаях разницы нет. Но давайте предположим обратное. Допустим, у нас есть таблица, состоящая из десяти полей, и мы хотим разделить ее на две. Пять полей в одну таблицу, пять в другую. В этом случае, чтобы сохранить целостность записей, таблицы надо связать по принципу «один-к-одному». С точки зрения быстродействия и экономии места мы вряд ли что-нибудь выиграем. Даже наоборот, придется создавать дополнительные кодовые поля. Зато теперь мы можем при необходимости заменять эти таблицы другими независимо друг от друга. Так обеспечивается принцип модульности данных: каждая таблица хранит данные из определенной предметной области, и в этом случае таблицы легче редактировать. Связь «один-к-одному» можно представить и наглядно. Допустим, у вас есть лист бумаги, на котором изображена большая таблица. Его можно разрезать по вертикали так, чтобы часть колонок попала на одну половину листа, а часть - на другую. Вы можете использовать половинки листа по отдельности. С ними могут работать одновременно два человека. Но в любой момент две части листа можно соединить, и таблица вновь станет целой.

Связь «один-к-одному» также возникает автоматически, если при задании поля в таблице воспользоваться типом данных «мастер подстановки».

Связь «многие-ко-многим»

Необходимость создания такой связи возникает, когда в обеих таблицах кодовые поля содержат повторяющиеся значения. Access не создает связи такого типа. Но их создание возможно следующим образом:

- Создайте две таблицы, которые необходимо связать отношением "многие ко многим";
- Создайте третью таблицу, называемую соединительной, и добавьте в нее поля с теми же определениями, что и поля первичных ключей в каждой из двух других таблиц. Поля первичных ключей соединительной таблицы служат внешними ключами. В соединительную таблицу, как и в любую другую, можно добавить и другие поля;
- Задайте первичный ключ этой таблицы таким образом, чтобы он включал в себя поля первичных ключей обеих основных таблиц. Для этого:
 - Откройте таблицу в режиме конструктора;
 - Выберите одно или несколько полей, которые необходимо определить в качестве первичного ключа. Чтобы выбрать одно поле, щелкните знак выбора строки для нужного поля;
 - Чтобы выбрать несколько полей, удерживайте нажатой клавишу CTRL и щелкните знак выбора строки для каждого из полей;
 - Нажмите кнопку *Ключевое поле* на панели инструментов.

4.5. Обеспечение целостности данных

При задании связи между двумя таблицами можно установить флажок *Обеспечение целостности данных*, чтобы Access продолжал следить за логической структурой записей в таблицах во время ввода данных и работы с базой. Конкретно это означает, что каждой записи подчиненной таблицы должна быть сопоставлена правильная запись базовой таблицы. Для того, чтобы этот флажок можно было установить, необходимо, чтобы типы данных связанных полей совпадали, а связь не была неопределенной. При установке этого флажка Access начинает следить за всеми изменениями, вносимыми в связанные поля.

Access не позволяет создать подчиненную запись, для которой не существует соответствующей записи в базовой таблице. Тем не менее, связанное поле в подчиненной таблице можно оставить пустым (это означает, что данная запись не имеет соответствующей).

Access не разрешает изменить значение в связанном поле базовой таблицы, если существуют связанные с ней записи в подчиненной таблице. Для того, чтобы это можно было делать, необходимо установить флажок *Каскадное обновление связанных полей* в окне диалога *Связи* - в этом случае изменение значения в связанном поле базовой таблицы приводит к аналогичным изменениям в соответствующих записях подчиненной таблицы, что гарантирует целостность данных в таблицах.

Access не разрешает удалить запись первичной таблицы, если существуют связанные с ней записи подчиненной таблицы. Однако, его можно разрешить, если установить флажок *Каскадное удаление связанных записей*. В этом случае удаление записи из базовой таблицы приводит к удалению всех связанных с ней записей подчиненной таблицы.

Access не разрешает изменять структуру данных в связанных полях (например, тип данных, размер поля и т.п.). Для того, чтобы произвести такие изменения, связь нужно предварительно удалить (и, при необходимости, восстановить после внесенных изменений).

4.6. Внесение изменений в схему связей

Чтобы внести изменения в параметры существующей связи, сделайте двойной щелчок на линии, изображающей связь в окне *Схема данных* (нужно только точно установить курсор мыши на линию связи). При этом снова открывается окно диалога *Связи*. То же самое можно сделать, если щелкнуть правой кнопкой мыши на линии и выбрать команду *Изменить связь*.

Для *удаления* существующей связи выделите соответствующую линию щелчком мыши и нажмите клавишу *Delete*. То же самое можно сделать, если щелкнуть правой кнопкой мыши на линии и выбрать команду *Удалить*.

После завершения работы со связями закройте окно диалога *Схема данных*. Если вы изменили макет окна, то есть состав включенных в него таблиц и их расположение, Access предложит сохранить его (при условии, что сами связи были сохранены ранее). Нажатие кнопки *Да* позволяет в дальнейшем воспользоваться тем же самым макетом.

Вопросы для самопроверки:

1. Для чего выполняется нормализация? Каковы ее основные принципы?
2. Из нижеприведенного списка выберите те типы полей, которые могут быть связаны:
 - а) счетчик и счетчик;
 - б) счетчик и числовое (целое);
 - в) денежное и числовое;
 - г) текстовое и счетчик;
 - д) счетчик и числовое (длинное целое).
3. В каком случае возникает связь «один-ко-многим»?
4. В каком случае возникает связь «один –к-одному»?
5. Какой тип связи возникает при связывании полей с помощью мастера подстановки?
6. В каких случаях удаление/изменение полей в базовой таблице позволяет удалять/изменять поля в подчиненной таблице?
7. При каком типе связи можно использовать *Обеспечение целостности данных* и *Каскадное обновление полей*?
8. Происходит ли автоматическое сохранение *Схемы данных* при ее изменениях?

5. МАСКИ ВВОДА ДАННЫХ

В *маске ввода* можно вводить любые символы и использовать специальные знаки. Специальные знаки и их интерпретация приведены в таблице.

Знак	Описание
0	Цифра (от 0 до 9, ввод обязателен; знаки плюс [+] и минус [-] не допускаются).
9	Цифра или пробел (ввод не обязателен; знаки плюс и минус не допускаются).
#	Цифра или пробел (ввод не обязателен; пустые знаки преобразуются в пробелы, допускаются знаки плюс и минус).
L	Буква (от A до Z или от A до Я, ввод обязателен).
?	Буква (от A до Z или от A до Я, ввод не обязателен).
A	Буква или цифра (ввод обязателен).
a	Буква или цифра (ввод необязателен).
&	Любой знак или пробел (ввод обязателен).
C	Любой знак или пробел (ввод необязателен).
.,:;- /	Десятичный разделитель и разделители тысяч, значений дат и времени. (Отображаемый знак зависит от настроек языка и стандартов на панели управления Microsoft Windows.)
<	Указывает перевод всех следующих знаков на нижний регистр.
>	Указывает перевод всех следующих знаков на верхний регистр.
!	Указывает заполнение маски ввода справа налево, а не слева направо. Заполнение маски знаками всегда происходит слева направо. Восклицательный знак в маске ввода можно помещать в любую позицию.
\	Указывает ввод любого следующего знака в качестве текстовой константы. Используется для отображения всех перечисленных в данной таблице знаков как текстовых констант (например, \A выводится как знак «А»).
Пароль	Значение <i>Пароль</i> , заданное для свойства <i>Маска ввода (InputMask)</i> , создает поле для ввода пароля. Любой знак, введенный в поле, сохраняется как знак, но отображается как звездочка (*).

Чтобы включить в маску текстовые константы, отличные от представленных в таблице, в том числе знаки и пробелы, следует просто ввести их в нужную позицию. Чтобы включить один из следующих знаков в качестве текстовой константы, необходимо перед ним ввести знак обратной косой черты (\).

Примеры масок ввода

В приведенной ниже таблице даны описания некоторых масок ввода и примеры значений, которые в них могут быть введены.

Описание маски ввода	Примеры значений
(000) 000-0000	(206) 555-0248
(999) 999-9999!	(206) 555-0248 () 555-0248
(000) AAA-AAAA	(206) 555-TELE
#999	-20 2000
>L???L?000L0	GREENGR339M3 МАЙ Р 452Ю7
>L0L 0L0	T2Ф 8M4
00000-9999	98115- 98115-3007
>L<??????????????	Мария Иван
ISBN 0-&&&&&&&-0	ISBN 1-55615-507-7 ISBN 0-13-964262-5
>LL00000-	DB51392-0493

Задания для самопроверки:

1. Создайте таблицу со следующей структурой. При задании типа данных используйте маску ввода (где это требуется).

Ф. И. О. студента	Домашний адрес	Телефон (домаш- ний, мобильный)	Электронный адрес
Иванов И. И.	61000, г. Харьков, ул. Иванова, 4	дом. 734-12-10 моб.(068)123-45-67	iii@rambler.ru
Петров П. П.	61000 г. Харьков, ул. Потемкина, д.5, кв. 18	дом. 684-35-690 моб.(097)987-65-43	petr@mail.ru

6. СОЗДАНИЕ ЗАПРОСОВ

6.1. Общие сведения

Одной из основополагающих частей любой СУБД является запрос. Запрос следовало бы даже признать самой важной ее частью, так как именно благодаря ему СУБД вообще существует и приносит практическую пользу. Таблицы в Access предназначены только для хранения данных. При этом они играют пассивную роль, т. е. сами не могут инициировать ни процесс запроса новых записей, ни передачу информации конечному потребителю. Для этого в СУБД служат другие объекты, и первый из них – запросы.

Запрос позволяет получить из одной или нескольких таблиц базы данных необходимую информацию, отвечающей заданному условию, выполнить (при необходимости) вычисления над данными. С помощью запроса можно обновлять данные в таблице, добавлять и удалять записи. Он может служить основой для формы или отчета.

Назначение запросов:

- выбор записей, отвечающих определенным критериям отбора без предварительного открытия конкретной таблицы или формы;
- выбор таблиц, содержащих нужные записи, с возможностью последующего добавления других таблиц;
- отбор полей, выводящихся на экран при отображении результирующего набора записей;
- создание новой таблицы на основе данных, полученных из существующих таблиц;
- обновление, добавление и удаление записей в таблицах;
- выполнение вычислений над значениями полей.

Те или иные запросы выполняются компьютером только тогда, когда это необходимо. Правда, медаль имеет и обратную сторону - исходя из своей идеологии, запрос сильно похож на моментальное фото, отражающее текущее состояние базы строго на момент выполнения данного запроса. Иными слова-

ми, если в данных произошли перемены после того, как запрос закончил свою работу, то указанные изменения в уже созданной выборке отражены не будут, так как виртуальная таблица с самими данными никакой связи не имеет. Это означает, что, во-первых, при проектировании любой базы данных следует обращать особое внимание на очередность запуска конкретных запросов, а во-вторых, если по каким-либо причинам пользователь нуждается в постоянном мониторинге в реальном режиме времени, то единственным способом достичь этого является повторение одного и того же запроса через короткие промежутки времени.

6.2. Виды запросов

По принципу своего действия все запросы Access делятся на несколько видов.

Запрос-выборка

Наиболее часто применяются запросы, производящие простую выборку некоторой информации из одной или нескольких взаимосвязанных таблиц. Результатом работы такого запроса является набор фактических данных, удовлетворяющих определенным, заранее заданным условиям. Полученные данные представляются в виде обычной таблицы, что позволяет создать на ее основе пользовательскую форму, представляющую результат на экране в удобном для восприятия виде. Кроме того, в отличие от обычной таблицы в запросе, кроме условий выборки, можно задавать дополнительные поля, в которых указывать формулы, оперирующие содержимым других полей.

После выполнения запроса СУБД создает виртуальную таблицу, в которую заносит выбранную информацию и хранит ее до тех пор, пока сгенерированная таблица не будет закрыта. Когда этот запрос закрывается, полученная таблица уничтожается, освобождая занимаемую память.

Запрос-действие

Вторым большим классом запросов является запрос-действие, в результате которого получается уже действительно самая обычная таблица. Упомяну-

тый выше запрос-выборка тоже генерирует таблицу, однако она существует только в оперативной памяти компьютера и исчезает сразу после прекращения работы запроса. В большинстве случаев этого оказывается вполне достаточно. Однако периодически возникает необходимость создать новую таблицу на основании данных из других таблиц. Например, по итогам семестра деканату нужно получить одну удобную таблицу, включающую в себя как обычные данные об успеваемости, так и расчетные величины (например, средний балл). Теоретически, ее можно соорудить вручную, что займет достаточно много времени и потребует приличных усилий от оператора. Практически же, можно сконструировать запрос-действие, который все возьмет на себя и уже через считанные секунды справится с задачей.

Во избежание совершения непреднамеренных действий с данными, которые нельзя отменить, разработчики СУБД Access настроили систему таким образом, что любой создаваемый запрос автоматически считается запросом-выборкой и работает в соответствии с этим. В том случае, когда пользователю или разработчику нужно получить запрос-действие, системе требуется об этом "сказать" отдельно.

Запрос-удаление

Третьей разновидностью запросов является запрос-удаление. Следует помнить, что Access обычным образом не позволяет удалять данные, занесенные в таблицу. Вообще-то, в ручном режиме пользователь может удалить одну или несколько строк, но это будет «топорная» работа. Хотя бы потому, что поле типа "счетчик" при таком удалении не обновляется, а, значит, по окончании процедуры последовательность будет нарушена. Следовательно, потом номер последней строки вовсе не свидетельствует о реальном количестве записей в таблице. Таким образом, как только из таблицы требуется что-либо удалить, следует воспользоваться запросом-удалением. В отличие от ручного режима, автоматическое удаление счетчик как раз обновляет.

6.3. Создание запроса – выборки

Наиболее часто используемым типом запроса является запрос на выборку. Например, из многостраничного железнодорожного расписания, вас могут интересовать только поезда, которые направляются в определенный город в заданный отрезок времени. Запросы на выборку можно также использовать для группировки записей и вычисления сумм, средних значений, подсчета записей и нахождения других типов итоговых значений. Внесенные в таблицу изменения автоматически отображаются в запросе.

Запрос-выборка производит вывод на экран любой информации, не меняя самой этой информации. То есть тот факт, что запрос "выбрал" конкретные данные и неким образом модифицировал их, вовсе не означает, что модификации подверглись исходные данные в самих таблицах.

Запрос-выборка просто генерирует ответ на заданный пользователем вопрос и на этом заканчивает свою работу. После выполнения запроса СУБД создает виртуальную таблицу, в которую заносит выбранную информацию и хранит ее до тех пор, пока сгенерированная таблица не будет закрыта. Когда этот запрос закрывается, полученная таблица уничтожается, освобождая занимаемую память.

Бланк запроса

В Access запросы удобно создавать в режиме *Конструктора*. Для создания запроса нужно выбрать в окне *База данных* вкладку *Запросы* и нажать кнопку *Создать*. В открывшемся окне *Новый запрос* выбрать *Конструктор*. Или можно сразу перейти к созданию нового запроса в режиме конструктора, выбрав *Создание запроса в режиме конструктора* на вкладке *Запросы*.

Окно *Конструктора запросов* разделено на две панели. Верхняя панель содержит схему данных запроса, которая включает выбранные для данного запроса таблицы. Таблицы представлены списками полей. Нижняя панель является бланком запроса, который нужно заполнить.

Бланк запроса представлен в виде таблицы. Такая таблица предназначена для конструирования структуры таблицы результата запроса и условий выборки записей. Первоначально эта таблица пуста. Каждый столбец бланка относится к одному полю, с которым нужно работать в запросе. Поля могут использоваться для включения их в таблицу запроса, для задания по ним сортировки, условий отбора записей, а также для выполнения вычислений.

При формировании столбца бланка запроса (поля таблицы запроса) необходимо знать следующее:

- в строку *Поле* включается имя поля, используемого в запросе;
- в строке *Сортировка* выбирается порядок сортировки записей результата запроса в строке (если нужно)
- в строке *Вывод на экран* отмечаются поля, которые должны быть включены в результирующую таблицу;
- в строке *Условие отбора* задаются условия отбора записей;
- в строке *или* задаются альтернативные условия отбора записей,

В ряде случаев в бланке запроса наряду с именем поля нужно отображать имя соответствующей таблицы, например, когда поля имеют одинаковые имена в разных таблицах. Если в бланке запроса строка *Имя таблицы* не отображается, нужно выполнить команду *Вид\Имена таблиц* или нажать соответствующую кнопку на панели конструктора запросов.

Поля бланка запроса

Каждый столбец бланка запроса соответствует одному из полей таблиц, на которых строится запрос. Кроме того, здесь может размещаться вычисляемое поле, значение которого вычисляется на основе значений других полей, или итоговое поле для групп записей, использующее одну из встроенных групповых функций Access.

Для включения нужных полей из таблиц в соответствующие столбцы запроса можно воспользоваться следующими приемами:

- в первой строке бланка запроса *Поле* щелчком мыши вызвать появление

кнопки списка и выбрать из списка нужное поле. Список содержит все поля таблиц, представленных в схеме данных запроса;

- перетащить нужное поле из списка полей таблицы в схеме данных запроса в первую строку бланка запроса;
- дважды щелкнуть на имени поля таблицы в схеме данных запроса;

Модификация запроса

Добавление таблицы в схему данных запроса осуществляется с помощью команды меню *Запрос\Добавление таблицы* или нажатием соответствующей кнопки панели *Конструктор запросов*. Команда добавления может быть выполнена также через контекстное меню, вызываемое на схеме данных запроса.

Добавление поля в бланк запроса осуществляется с помощью одного из действий, рассмотренных выше, например, перетаскиванием мышью имени поля из таблицы в схеме данных в нужное место бланка. Все столбцы полей справа от него передвинутся на один столбец вправо.

Удаление поля в бланке запроса требует предварительного выделения соответствующего столбца. Для этого нужно переместить курсор в область маркировки столбца, где он примет вид черной стрелки, направленной вниз, и щелкнуть кнопкой мыши. Далее необходимо нажать клавишу *Delete*.

Для перемещения поля в бланке запроса надо выделить соответствующий столбец и перетащить его в новую позицию с помощью мыши. Столбец, на место которого перемещен новый, и все столбцы справа от него будут сдвинуты вправо.

6.4. Условия отбора записей

Условия отбора записей могут задаваться для одного или нескольких полей в строке *Условие отбора* бланка запроса. Для описания условий отбора записей, используемых в запросе, используются выражения.

Выражения в Access

Под выражением подразумевается любая комбинация операторов, констант, значений текстовых констант, функций, имен полей (столбцов), элементов управления или свойств, результатом которой является *конкретное значение*.

Каждое выражение может содержать один или несколько операторов и одну или несколько констант, идентификаторов или функций. Выражение может быть сколь угодно сложным.

Константы – характеризуют неизменные значения. Их часто используют для создания значений по умолчанию и для сравнения значений в полях таблиц. Значения констант определяются пользователем при вводе выражения

Идентификаторы – это имена объектов в Access (например, полей таблиц или запросов), которые при вычислении выражений заменяются их текущими значениями (для определения этой операции обычно используется термин *возвращают*)

Функции возвращают в выражение значение вместо имени функции. В отличие от идентификаторов, большинство функций требуют заключать в скобки свои аргументы — идентификаторы или значения подвыражений. Например, функция Date(), имеющая пустой список аргументов, возвращает текущую дату.

Используемые в операции константы, идентификаторы и функции называются *операндами*.

Операторы – обычные значки арифметических операций (+, -, *, /) и другие символы и аббревиатуры. Оператор – это символ или слово, например. = или Or, указывающие выполнение операции над одним или несколькими элементами. Операторы позволяют выполнять операции над элементами формулы. Access позволяет использовать различные типы операторов для вычислений:

- арифметические операторы – служат для выполнения арифметических операций над числами;
- операторы сравнения – используются для сравнения двух значений. Результатом сравнения может являться логическое значение: либо *ИСТИНА*, либо *ЛОЖЬ*;
- текстовый оператор *конкатенации* & – используется для объединения нескольких текстовых строк в одну строку;

Запрос записей в определенном диапазоне значений может выполняться с использованием следующих операторов сравнения:

= равно;

> – больше, чем;

< – меньше, чем;

<> – не равно;

>= – больше или равно;

<= – меньше или равно.

Допустим, нас интересуют данные анализа качества атмосферного воздуха, сделанные после 1 января 2007 г. В строку Условия отбора для поля *Дата анализа* нужно ввести >1.01.07

Как правило, операторы сравнения используются в числовых полях или полях дат, но они могут применяться и в текстовых полях.

Если вы не уверены в написании какого-либо слова, то можете использовать оператор LIKE и подстановочный знак. Напомним, что знак вопроса заменяет один символ, а звездочка * – группу символов. Например, выражение: LIKE П* выполняет поиск слов, начинающихся с буквы П. Выражение: LIKE П?Л выполняет поиск слов у которых первая и третья буква соответственно П и Л, а вторая буква может быть любой (пел, пал, пол и т.п).

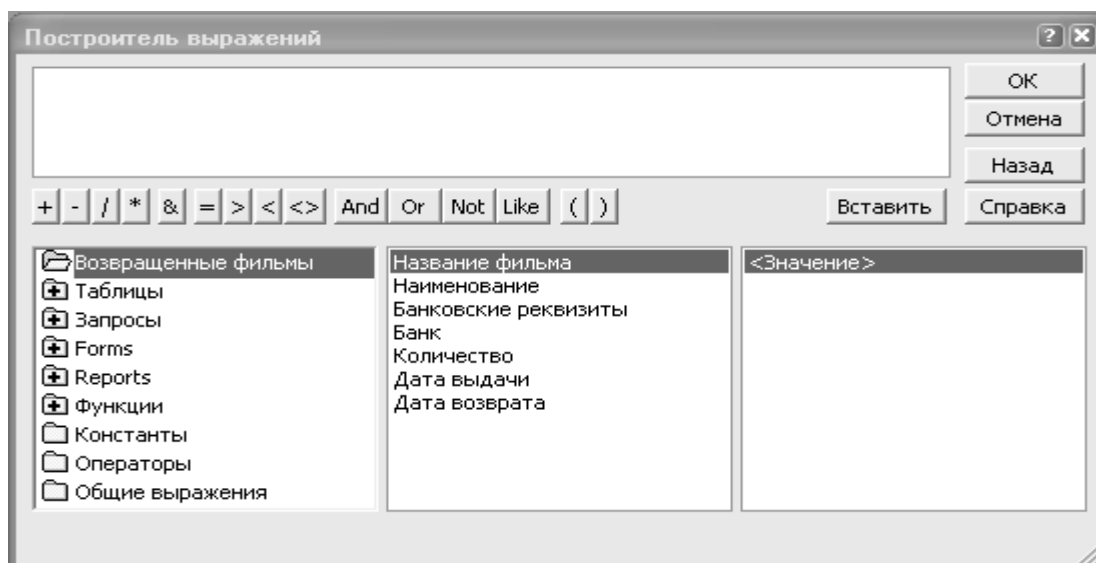
6.5. Построитель выражений

Для упрощения процесса добавления в запрос функций, констант, операторов можно использовать *Построитель выражений*.

Для вызова *Построителя выражений* достаточно перевести любой запрос в режим конструктора, попасть маркером мыши в любое поле в нижней части бланка и нажатием правой кнопки мыши вызвать контекстное меню. Среди всего прочего в нем есть пиктограмма волшебной палочки и режим *Построить*. Активизация этого режима приводит к появлению на экране рабочего окна построителя выражений, в котором и предстоит работать до получения необходимого результата. Только не следует забывать, что сгенерированное

выражение программа поместит в ту ячейку, в которой находился маркер мыши на момент активизации окна мастера.

Построитель выражений состоит из четырех рабочих областей.



Верхняя область, самая большая, является как бы листом для записи результата по мере его формирования. В любой момент разработчик может щелкнуть по ней мышью и в ручном режиме произвести любые коррективы, какие только посчитает нужными.

Три нижних области выполняют функции меню, в которых из предлагаемых вариантов пользователю предстоит выбирать то или иное значение. Нижняя левая область "отвечает" за общие разделы. Работая на манер *Проводника*, она позволяет "сказать", что конкретно вас в данный момент интересует: таблицы, запросы, формы, определенная группа функций или метод обмена данными с внешними приложениями. Нижняя средняя область показывает, какие подразделы существуют в рамках выделенного раздела. Если разработчик поместил выделение в левой области на пиктограмму папки *Таблицы* и "раскрыл" ее (делается точно так же, как и в *Проводнике*), то программа в качестве содержимого папки тут же показывает наименования всех таблиц, какие только существуют в данной БД. Выбор конкретной таблицы, в свою очередь, приводит к появлению в центральной области перечня полей, которые заданы в данной таблице. Соответственно, выбрав мышью конкретное поле в средней области, вы, тем самым, даете построителю выражений понять, что желаете поместить в

формулу или выражение ссылку на содержание конкретного поля конкретной таблицы. Когда описанные манипуляции выполнены, самое время "сказать" программе, что данную величину требуется поместить в запись выражения в верхней области. Делается это либо нажатием клавиши *Ввод* на клавиатуре, либо двойным щелчком мыши, либо нажатием экранной кнопки *Вставить*, расположенной на окне построителя выражений. В случае ошибки всегда можно отменить последнюю операцию (кнопка *Назад*) или даже отменить генерацию всего выражения в целом (кнопка *Отмена*). Только не перепутайте, *Назад* означает назад на один шаг, а *Отмена* означает отменить сам построитель выражений.

Для удобства работы, наиболее часто используемые элементы выражений вынесены авторами Access в виде отдельных кнопок непосредственно на само окно построителя. Это такие элементы, как скобки, обозначения операций сложения, умножения и тому подобные. Остальное расположено в соответствующих папках. Всего построитель выражений имеет восемь базовых папок: таблицы, запросы, формы, отчеты, функции, константы, операторы и общие выражения. Назначение первых четырех категорий совершенно очевидно, а вот последующие имеет смысл рассмотреть поподробнее. Папка *Функции* содержит в себе два раздела: встроенные функции и функции определенные в текущей базе данных. Встроенные функции открывают доступ ко всему богатству стандартных функций Access данной версии. По умолчанию программа показывает их все сразу (в средней и правой областях), но при желании разработчик может выделить лишь конкретные темы (например, только функции для операции над датой/временем). В этом случае в правой области Access показывает только те конкретные функции, которые относятся к выбранной области. Если выделить какую-либо из этих функций, то в самой нижней части окна построителя выражений появляется подсказка, показывающая формат написания данной функции и перечень информации, необходимой для ее работы. Аналогичным образом СУБД позволяет пользоваться и локальными функциями собственной разработки, заданными в рамках текущей базы данных. Константы, соответствен-

но, охватывают величины, неизменные в рамках СУБД. Это "истина", "ложь", "пустая строка" и "пустое значение". Операторы же описывают все операции, какие только в Access разрешены над данными (арифметические, логические, сравнения). Раздел *Общие выражения* необходим, как правило, для украшательских целей, так как он содержит конструкции стандартные, применяющиеся в основном для стилистического оформления результата работы, например такие, как ["Страница "&Page], что приводит к появлению на листе бумаги при печати конструкции "Страница "номер" где, вместо слова "номер" стоит текущий номер страницы.

Особенностью работы построителя выражений является то, что сам построитель совершенно не следит за корректностью составленного выражения. Это значит, что даже при помощи построителя можно соорудить нечто неработоспособное. Правда, построитель в меру своих сил пытается наметать, что в тех или иных местах нужно самостоятельно вставить какое-нибудь определенное содержимое. Например, между двумя функциями программа вставляет надпись "Выражение", показывая, что в этом месте нужно вставить какое-либо конкретное выражение. То есть использование построителя выражений значительно облегчает создание приложений, но не снимает с человека ответственности за конечный результат.

6.6. Вычисляемые поля

В запросе для каждой записи могут производиться вычисления с числовыми и строковыми значениями или значениями дат с использованием данных из одного или нескольких полей. Результат вычисления образует в таблице запроса новое вычисляемое поле. В исходных таблицах базы данных новых полей не создается. При каждом выполнении запроса производится вычисление с использованием текущих значений полей.

В выражениях вычисляемых полей помимо имен полей могут использоваться функции. В результате обработки выражения может получаться только одно значение.

Выражение вводится в бланк запроса в пустую ячейку строки *Поле*. Имя вычисляемого поля, стоящее перед выражением, отделяется от него двоеточием. Например, вводится вычисляемое поле *Сумма* как

Сумма: [Цена][Количество],*

где *Цена* и *Количество* — имена полей.

Имя вычисляемого поля *Сумма* становится заголовком столбца в таблице с результатами выполнения запроса. Это имя можно изменить. Для вычисляемых полей, как и для любых других, допускаются сортировка, задание условий отбора и расчет итоговых значений.

6.7. Запросы с параметром

Конкретное значение в выражение условия отбора может вводиться непосредственно в бланк запроса. Однако при решении практических задач значительно удобнее вводить значение в диалоге с пользователем в процессе выполнения запроса. Для того чтобы выводилось диалоговое окно, обеспечивающее ввод конкретного значения в условия отбора, нужно определить параметр запроса.

Имя параметра запроса может задаваться непосредственно в строке *Условия отбора* в квадратных скобках. При выполнении запроса это имя появится в диалоговом окне *Введите значение параметра*. Наиболее часто для задания значения параметра запроса используют формы.

6.8. Многотабличные запросы

Многотабличный запрос позволяет сформировать записи результата путем объединения взаимосвязанных записей из таблиц базы данных и выбора из них нужных полей и записей. Многотабличный запрос часто осуществляет объединение данных, которые на этапе проектирования были разделены на множество объектов в соответствии требованиями нормализации. В результате выполнения запроса формируется таблица с повторяющимися данными, в которой каждая запись собирает необходимые данные из разных таблиц.

6.9.Просмотр результата запроса

Внешне отображение результатов запроса не отличается от отображения таблицы. С ней можно работать так же, как и с обычной таблицей, в частности не только просматривать, но и изменять данные. Изменения сохраняются в записях базовой таблицы, на основе которой построен запрос. Если запрос создан на основе двух и более связанных таблиц, то корректировка возможна не всегда

Условия, при которых можно добавить или обновить записи в запросе:

- таблица является единственной в запросе;
- таблицы в запросе связаны отношением "один-к-одному";
- если таблицы в запросе связаны отношением "один-ко-многим", можно изменять поля только в таблице "многие".

Вопросы для самопроверки:

- 1.Для чего в СУБД используются запросы?
- 2.Что является результатом работы: запроса-выборки, запроса-действия, запроса-удаления?
3. Как долго хранится в памяти компьютера таблица, сформированная в результате работы запроса?
4. Опишите порядок создания запроса-выборки
5. Каким образом можно изменять уже созданный запрос?
6. Перечислите и опишите выражения в Access.
7. Чем отличаются операнды от операторов?
8. Опишите принципы работы построителя выражений
9. Возможно ли проводить вычисления с помощью запроса? Если да, то как это делается?
10. Опишите порядок создания запроса с параметром.

7. ФОРМЫ

7.1. Назначение и виды форм

Формы предназначены для взаимодействия с пользователем, представляя удобный способ ввода и вывода информации (просмотровые формы), а также для управления БД (кнопочные формы).

Обычно работа с БД начинается с кнопочной *Главной экранной формы*, на которой располагаются кнопки управления, позволяющие выполнять все операции в БД. Часто функции просмотровых и кнопочных форм совмещаются в одной форме. Просмотровые формы имеют источник информации (таблица или запросы). Если источников информации несколько, они должны быть предварительно связаны между собой.

Все изменения, вносимые с помощью формы, автоматически переносятся в связанные с ней таблицы. Однако, изменения в структуре таблиц и запросов (например, добавление или удаление поле) автоматически не вносятся в форму и форма должна быть изменена вручную.

Формы можно создавать с помощью *Мастера форм* или в режиме *Конструктора*. Более удобно пользоваться *Мастером форм*. Созданную форму можно потом видоизменить в режиме *Конструктора*. *Мастер форм* позволяет создавать формы нескольких видов:

- В один столбец – каждая запись располагается на отдельной странице, каждое поле – в отдельной строке. Просмотр данных производится перелистыванием с помощью счетчика. Обычно используется для ввода данных.
- Ленточная – на странице помещается несколько записей, одна под другой. Обычно используется для просмотра и редактирования данных.
- Табличная – имеет вид обычной таблицы. Обычно используется для просмотра и редактирования данных.
- Выровненная - каждая запись располагается на отдельной странице, но поля размещаются в линию. Обычно используется для ввода данных.

7.2. Редактирование формы

Построенная с помощью *Мастера* форма обычно требует доработки в режиме *Конструктора*. В частности, элементы управления на форме можно задавать только в режиме *Конструктора*. В этом режиме можно также изменять размеры объектов, их размещение, цветовую гамму, шрифты и т.д. Производится редактирование командой *Свойства* контекстного меню или с помощью кнопок на панели инструментов. Надо помнить, что при вызове контекстного меню для редактирования конкретного элемента формы указатель мыши предварительно нужно поместить на этот элемент, для редактирования общих свойств формы (подписи, разделительных линий, полос прокрутки и т.д.) указатель мыши помещается на свободное место формы.

Важной задачей редактирования формы является размещение на ней элементов управления. Размещение элементов управления осуществляется с помощью панели инструментов. На этой панели располагаются следующие инструменты для создания управляющих элементов:

- *Переключатель* – кнопка для выбора одного из взаимоисключающих параметров.
- *Выключатель* – кнопка, находящаяся только в двух состояниях: включен - выключен.
- *Список* – формирует список. Список всегда раскрыт. Для просмотра списка используются полосы прокрутки.
- *Поле со списком* – формирует поле ввода с раскрывающимся списком. Ввод значения в поле производится выбором элемента из списка.
- *Кнопка* – служит для выполнения определенных действий. Эти действия могут задаваться макросами.
- *Флажок* – кнопка для выбора не исключаящих друг друга параметров.
- *Рисунок* – кнопка для вставки рисунка из файла
- *Свободная рамка объекта* – формирует рамку внедренного объекта.
- *Присоединенная рамка объекта* - формирует рамку объекта, сохраненного в БД.

- *Разрыв страницы* – формирует многостраничные формы
- *Вкладка* – создает форму с несколькими вкладками.
- *Подчиненная форма/отчет* – создает подчиненную форму или отчет
- *Диаграмма* – создает диаграмму

Кроме элементов управления на панели инструментов находятся следующие объекты:

- *Надпись* – для создания подписи к другим элементам формы
- *Поле* – добавляет в форму текстовое поле. Это поле можно связать с полем таблицы.
- *Линия* – рисует линию. С помощью палитры задаются толщина и вид линии.
- *Прямоугольник* – рисует прямоугольник.

Элементы управления в режиме Конструктора формы обычно создаются с помощью *Мастеров*. Для активизации *Мастеров* перед созданием элементов управления необходимо щелкнуть по элементу *Мастера* на панели элементов. Например, для создания кнопки необходимо щелкнуть по элементам *Мастера* и *Кнопка* на панели элементов, затем щелкнуть в том месте формы, где должна быть размещена эта кнопка. Появляется диалоговое окно *Создание кнопок*, с помощью которого с кнопкой связывают некоторое действие (их еще называют событиями). Аналогично создаются другие элементы управления: включатели, переключатели, флажки и т.д.

Вопросы для самопроверки:

1. Для чего служат формы?
2. Какие виды форм вы знаете?
3. Перечислите способы создания форм.
4. Возможно ли добавление и удаление полей в уже созданной форме? Каким образом это выполняется?
5. Опишите последовательность действий при: а) создании кнопки открытия формы; б) создании кнопки закрытия формы; в) добавлении нового поля; г) добавлении поля со списком; д) изменении шрифта и цвета надписей.

8. ОТЧЕТЫ

8.1. Назначение и создание отчетов

Отчеты позволяют выбрать из базы данных требуемую информацию и *оформить ее в виде документов, которые можно просмотреть и напечатать*. Источником данных для отчета может быть таблица или запрос. Кроме данных, полученных из таблиц, в отчете могут отображаться вычисленные по исходным данным значения, например итоговые суммы.

Отчеты Access имеют много общего с формами. Однако, в отличие от форм, отчеты не предназначены для ввода и правки данных в таблицах. Они позволяют лишь просматривать и печатать данные. В отчете невозможно изменить исходные данные с помощью элементов управления, как это можно сделать в формах

Отчет может создаваться с помощью *Мастера отчетов* или в режиме *Конструктора отчетов*. Обычно удобно сначала использовать *Мастер отчетов*, а затем отчет, созданный *Мастером*, доработать в режиме *Конструктора*.

8.2. Разделы отчета

Разделы отчета видны в режиме Конструктора.

В *Заголовок отчета* помещается шапка отчета. В *Верхний* и *Нижний колонтитул* помещают надписи, повторяющиеся на каждой странице отчета. Это могут быть, например, заголовки столбцов отчета, номера страниц и даты. При определении содержания этих разделов следует исходить из требований к оформлению отдельных страниц отчета. В *Области данных* размещаются поля таблиц базы данных или запросов.

При необходимости группировки записей по полю в окно *Конструктора отчетов* могут быть добавлены разделы *Заголовок группы* и *Примечание группы*. В *Заголовке группы*, как правило, размещаются поля, по которым производится группировка. В *Примечании группы* могут быть размещены выражения для подведения итогов по группе. Допускается до 10 уровней группировки выводимых записей.

Вопросы для самопроверки:

1. Для чего предназначены запросы?
3. В чем заключается отличие отчетов от форм?
4. Какие части отчетов являются обязательными?
5. Какие части отчетов могут в них не включаться?

СОДЕРЖАТЕЛЬНЫЙ МОДУЛЬ 2.2.

СОЗДАНИЕ ПРЕЗЕНТАЦИЙ В MS POWER POINT

9. НАЗНАЧЕНИЕ И СТРУКТУРА ПРЕЗЕНТАЦИИ MS POWER POINT

9.1. Общие сведения

Power Point создает файл презентации, который имеет расширение PPT.

Презентация - это набор слайдов, объединенных возможностью перехода от одного слайда к другому и хранящихся в общем файле.

Слайд - логически автономная информационная структура, содержащая различные объекты, которые представляются на общем экране монитора, листе бумаги или на листе цветной пленки в виде единой композиции. Слайд состоит из набора областей.

В составе слайда могут присутствовать следующие области: текстовые области, графические области (рисунки), таблицы, диаграммы, организационные диаграммы, мультимедийные области (звук, видео), области объектов OLE. В структуре слайда различают заголовок, основную область, фон, колонтитул.

9.2. Структура окна Power Point

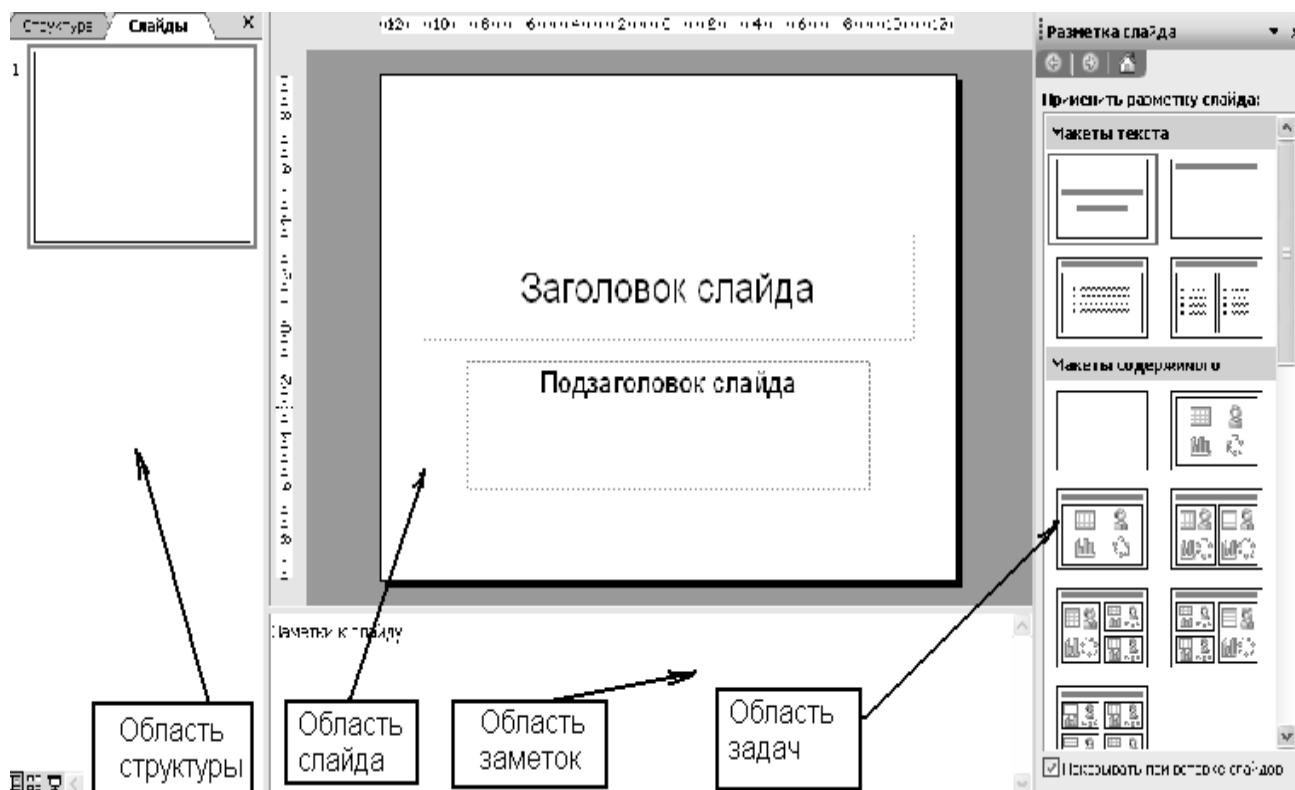
Окно презентации Power Point состоит из следующих элементов:

Область структуры презентации. Позволяет изменять текст и порядок следования слайдов, а также удалять и добавлять новые слайды в презентации

Область слайда. Позволяет изменять текст, графические изображения, макет и дизайн каждого слайда.

Область заметок. В ней отображаются заметки, которые добавлены к слайдам

Область задач. В ней отображаются команды, необходимые для создания и внесения изменений в презентацию



Размеры любой из областей можно изменить, перетаскив ее границу

Вопросы для самопроверки:

1. Для чего используется презентация MS POWER POINT?
2. Перечислите элементы, из которых состоит слайд.
3. Из каких областей состоит окно презентации MS POWER POINT и для чего они предназначены?

10. ПОСЛЕДОВАТЕЛЬНОСТЬ РАБОТЫ НАД ПРЕЗЕНТАЦИЕЙ

PowerPoint предоставляет пользователю большое количество шаблонов презентаций на различные темы. Такие шаблоны содержат слайды, оформленные определенным образом. В поле слайда размещаются местозаполнители, в которые можно вставить свой текст, графику, а также таблицы или диаграммы. Достаточно часто местозаполнители содержат рекомендации по смысловому наполнению данной области. При желании, пользователь может изменить художественное оформление любого шаблона презентации, выбрав дизайн по своему вкусу.

Если нужный шаблон презентации отсутствует или не удовлетворяет пользователя, то он может создать презентацию самостоятельно «с нуля».

Весь процесс разработки презентации может быть разбит на несколько этапов:

1. Разработка структуры презентации
2. Создание отдельных слайдов
3. Составление связанной последовательности слайдов
4. Создание вспомогательной поддержки презентации
5. Планирование демонстрации

11. РЕЖИМЫ РАБОТЫ В POWER POINT

В соответствии с разделением работы над презентацией на этапы в PowerPoint предусмотрены различные режимы работы:

Режим структуры. В режиме структуры PowerPoint отображает текстовое содержимое презентации, позволяя вводить новый текст или редактировать существующий (подобно любому текстовому редактору), не отвлекаясь при этом на детали оформления, присутствующие в других режимах. Поскольку работа идет только над текстом, пользователь может целиком сконцентрироваться на смысловом наполнении слайдов и на словесном изложении своих идей. Этот режим прекрасно приспособлен для выбора оптимальной последовательности подачи материала, а дизайн и другие элементы оформления внешнего вида презентации прорабатываются после переключения в другие режимы.

Режим слайда. В этом режиме удобно конструировать или корректировать отдельные слайды презентации. Этот режим позволяет вводить и редактировать текст, вставлять графические объекты, диаграммы (графики) и таблицы, мультимедийные объекты. Кроме того, слайд можно дополнить собственным фоном и текстовыми комментариями.

Режим сортировщика слайдов. В этом режиме миниатюрные копии слайдов всей презентационной последовательности располагаются равномерными рядами в окне просмотра. Пользователь может проследить влияние любого общего параметра, такого как оформление фона слайдов или выбор цветовой схемы, на вид презентации в целом. В этом режиме отсутствует возможность изменения содержания конкретных слайдов, зато доступны средства их удаления, дублирования и изменения порядка следования. С помощью данного режима можно перед печатью или демонстрацией презентации проверить ее на наличие противоречивых слайдов или грубых ошибок в представлении информации. Режим сортировщика слайдов также используется для задания и изменения переходных эффектов, реализующихся на экране при смене слайдов во время демонстрации презентации.

Режим заметок. Предназначен для создания страниц заметок, которые могут использоваться докладчиком во время презентации или служить в качестве раздаточного материала. В этом режиме уменьшенная копия слайда располагается в верхней части страницы, тогда как нижнюю, свободную часть страницы, можно заполнить текстовым материалом, содержащим пояснения к данному слайду. Выбрав подходящий масштаб отображения, можно одновременно видеть на экране уменьшенный вариант слайда и иметь возможность вводить сопроводительный текст.

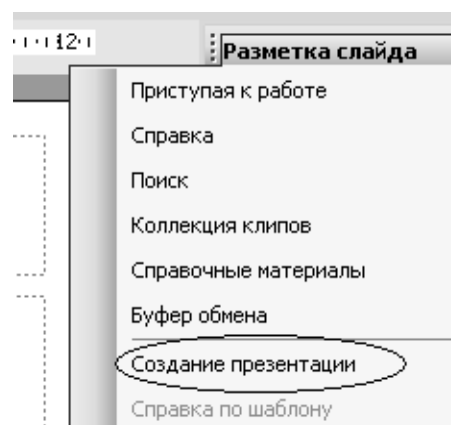
Режим показа слайдов. Позволяет демонстрировать слайд или всю презентацию в динамике, слайд за слайдом. Демонстрация выполняется точно так, как будет осуществляться показ готовой презентации - с использованием переходов, анимации, специальных эффектов и переходов от одного слайда к другому.

Вопросы для самопроверки:

1. Перечислите режимы работы над презентацией PowerPoint
2. Какие действия можно выполнять в каждом из режимов работы?

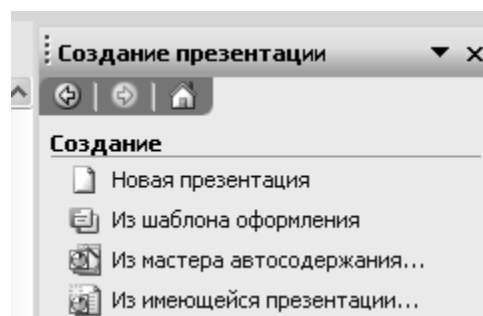
12. СОЗДАНИЕ ПРЕЗЕНТАЦИИ

Доступ к шаблонам осуществляется с помощью режима *Создание презентации Области задач* Power Point. Перейти к этому режиму можно либо с помощью меню *Файл/Создать* либо выбрав элемент *Создание презентации* в выпадающем меню *Области задач*.



В области *Создание* перечислены возможные варианты создания новой презентации:

- *Новая презентация без какого-либо форматирования.* Это режим «по умолчанию», который автоматически задается при открытии приложения Power Point.
- *Из шаблона оформления.* В этом режиме можно выбрать один из существующих шаблонов оформления слайдов. Шаблон оформления представляет собой профессионально оформленный слайд, включающий фоновый рисунок, цветовую схему и шрифты, к которому остается только добавить текст.
- *Из Мастера автосодержания.* В этом режиме, кроме оформления, пользователю предлагается заранее заданная структура содержания на некоторую тему.

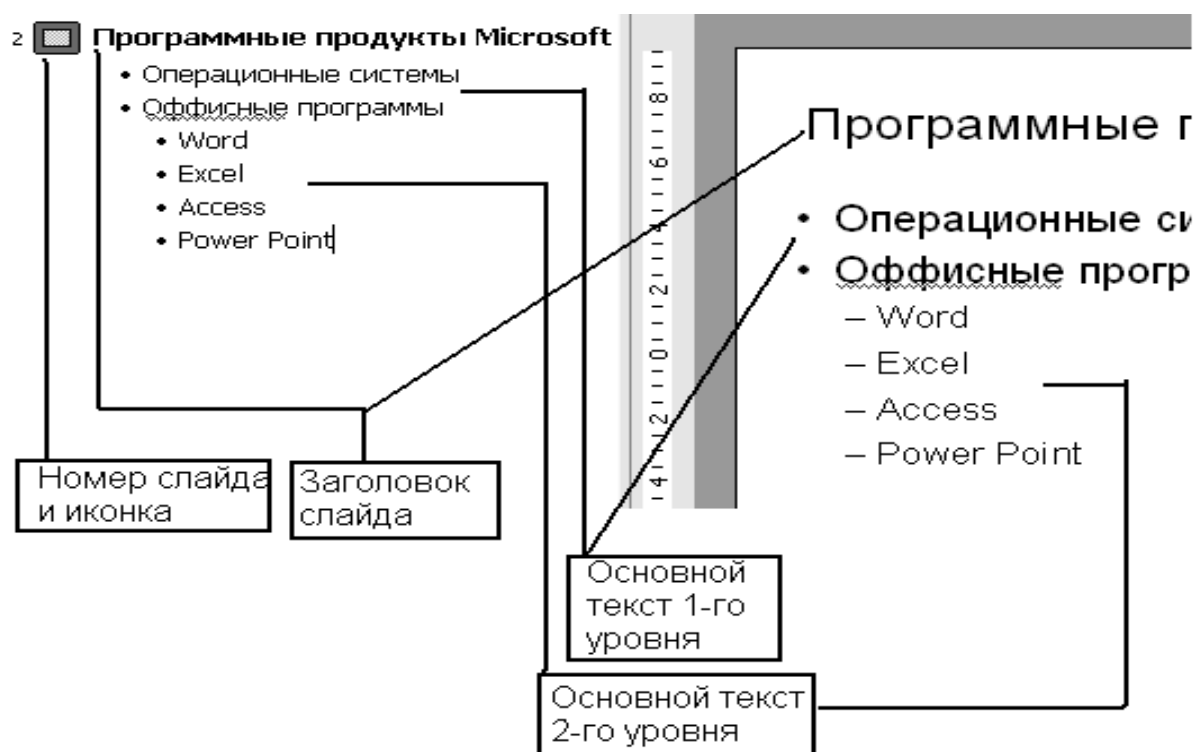


12.1. Создание презентации в режиме структуры

Этот режим в основном предназначен для работы над текстом. Он позволяет сконцентрироваться на словесном изложении идей, подобрать оптимальную последовательность подачи материала. Этот режим позволяет легко перемещать текст между слайдами.

Ключевым понятием режима структуры является *уровень текста*. Текст самого высокого уровня воспринимается как заголовок, а последующие - как основной текст. Режим структуры допускает создание до пяти уровней основного текста, каждый из которых располагается правее предыдущего и имеет меньший размер шрифта.

Для повышения уровня используется кнопка *Уменьшить отступ* на панели инструментов *Форматирование*, а для понижения уровня - кнопка *Увеличить отступ*.



12.2. Создание отдельного слайда

Для создания слайда используется меню *Вставка/Создать слайд*. При этом в области задач появляется закладка *Разметка слайда* с набором макетов слайда, с помощью которых могут быть созданы слайды, содержащие кроме текста изображения, графики, таблицы, маркированные списки, организационные диаграммы, а также их сочетания. Однако, можно начать с пустого слайда (макет "Пустой слайд") и самостоятельно сформировать свою структуру слайда.

12.3. Размещение объектов на слайде

Слайд представляет собой многослойную структуру, в которой количество слоев отвечает количеству объектов. Самым глубоким слоем является фон слайда. Другие объекты слайда могут произвольно размещаться по слоям и, при необходимости, перемещаться. В зависимости от этого будет определяться их видимость на слайде (объект может быть загорожен другим объектом, расположенным на переднем слое). В любой момент времени можно менять расположение объектов, как в координатах плоскости, так и по глубине слоев.

12.4. Работа с текстом

Текст - один из основных элементов любого слайда. Его можно ввести через клавиатуру или импортировать из другого приложения, например из Word. Создание текстового элемента и последующее изменение выполняется почти так же, как это делается в Word.

В PowerPoint обеспечиваются следующие возможности работы с текстом: задание шрифта, начертания, размера, цвета и специальных параметров; задание интервалов между строками; задание отступов; выравнивание текста; вращение текста; маркирование текста; использование позиций табуляции; оформление текста рамкой; применение к текстовому элементу эффектов для графики.

Для того, чтобы находящаяся на слайде текстовая информация была хорошо читаема при демонстрации необходимо придерживаться правила: размер шрифта заголовка слайда - не менее 40 пт.; размер текста слайда - не менее 32 пт. (в таблицах - не менее 28 пт.)

12.5. Анимация объектов слайда

В ходе демонстрации слайда абзацы маркированных списков, графические и текстовые элементы могут появляться все сразу, либо последовательно. Это зависит от того, назначен ли объекту эффект анимации. Все не анимацион-

ные объекты появляются на слайде одновременно при его открытии. Затем последовательно появляются объекты, для которых установлен эффект анимации.

Создание анимации на слайде включает два шага:

- анимация и звуковое сопровождение отдельного объекта
- установление последовательности появления объектов на слайде.

12.6. Построение последовательности слайдов

Работа в режиме сортировщика

Всю последовательность слайдов можно наблюдать в режиме сортировщика слайдов. В этом режиме нет возможности изменять отдельный слайд, но удобно производить следующие действия:

- сортировка слайдов в презентации;
- копирование слайдов в презентации;
- назначение эффектов перехода от слайда к слайду;
- добавление итогового слайда;
- изменение общего дизайна презентации;
- хронометраж слайд-фильма.

Сортировка слайдов. Сортировка слайдов осуществляется перетаскиванием слайдов на нужное место в режиме сортировщика слайдов.

Копирование слайдов. Копирование слайдов выполняется путем выделения копируемых слайдов и использованием меню Правка/Дублировать. Также можно скопировать слайд из другой презентации (другого файла). Для копирования слайдов из другой презентации в PowerPoint используется меню Вставка/Слайды из файлов.

Определение эффекта смены слайдов. Назначение эффекта перехода в режиме сортировщика осуществляется путем выбора нужного эффекта через меню Показ слайдов/Смена слайда либо через закладку Смена слайда Области задач.

Итоговый слайд - это слайд, определенный для группы слайдов или всей презентации, на котором в автоматическом режиме собираются все заголовки слайдов группы в виде маркированного списка. Для его создания:

1. В режиме *Сортировщика слайдов* выделить необходимые слайды
2. На панели инструментов *Сортировщик слайдов* нажать кнопку *Итоговый слайд*. Перед первым выделенным слайдом появится новый слайд с маркированным списком, состоящим из заголовком выделенных слайдов.

Изменение дизайна слайдов. В режиме сортировщика слайдов можно эффективно обрабатывать группы слайдов. Процедуры обработки нескольких слайдов сводятся к выделению группы рядом стоящих слайдов и применению к ней требуемых действий.

Например, чтобы изменить фон в группе слайдов, следует выделить эту группу и воспользоваться меню *Фон*, определяя требуемые параметры фона. Точно также можно изменить для отдельных слайдов цветовую схему.

Работая в режиме сортировщика можно воспользоваться кнопкой "Формат по образцу" для того, что бы копировать цветовую схему одного слайда в другой или в целую группу слайдов.

12.7. Шаблон презентации

Самый простой способ определения внешнего вида презентации - это использование шаблона (меню *Формат/Шаблон оформления*). Шаблон задает фон, цветовую схему и набор шрифтов текста. Каждым из этих элементов дизайна можно управлять индивидуально, не применяя новый шаблон. Существует возможность изменения дизайна отдельного слайда или группы слайдов, не меняя другие слайды.

Также пользователи имеют возможность создать свой шаблон. Делается это путем разработки базовых элементов презентации и сохранения их в виде файла-шаблона.

Цветовая схема презентации

Одним из компонентов шаблона является цветовая схема, которая определяет упорядоченную последовательность из восьми цветов, предназначенных для отображения элементов оформления слайда. На слайде может быть восемь полей, цвет которых формирует цветовую схему и определяет цветовое оформление элементов слайда:

Фон - цвет, используемый в качестве фона слайда. На отдельных слайдах цвет фона может перекрываться другим цветом фона, указанным пользователем.

Текст и линии - цвет текста в маркированных списках и текстовых блоках. Этот же цвет используется для линий и стрелок, нарисованных с помощью инструментов *Линия* и *Стрелка*, для контуров автофигур и других объектов, нарисованных с помощью инструментов рисования.

Тени - цвет теней.

Заголовок - цвет заголовков и подзаголовков слайдов.

Заливка - цвет заливки автофигур и объектов, созданных с помощью инструментов рисования. Этот же цвет используется для заливки первого ряда данных на диаграммах.

Акцент - цвет, используемый в качестве второго цвета на диаграммах, организационных диаграммах и других внедренных объектах.

Акцент с гиперссылкой - цвет, используемый в качестве третьего цвета на диаграммах, организационных диаграммах и других внедренных объектах. Этот же цвет применяется для объектов-носителей гиперссылок (объектов, инициирующих переходы к другим слайдам, презентациям или Web-страницам).

Акцент с последующей гиперссылкой - цвет, используемый в качестве четвертого цвета на диаграммах, организационных диаграммах и других вставленных объектах. Он также применяется для активизированных объектов-носителей гиперссылок.

12.8. Мастер-слайд

В любой презентации существует специальный слайд, который не участвует в демонстрации, а используется для определения и хранения образца для других слайдов. Этот слайд называется мастер-слайдом.

Размещенные на мастер-слайде объекты будут присутствовать на всех остальных слайдах.

Это позволяет автоматизировать такие рутинные работы, как размещение колонтитула, даты, общего фона и т.д. Однако к объектам слайда, которые пришли с мастер-слайда, не могут быть применены эффекты анимации. Вызвать мастер-слайд можно зажав клавишу *Shift* и щелкнув по кнопке *Обычный режим* в нижней части *Области структуры* окна Power Point. В оформление мастер-слайда можно внести изменения. Внесенные в мастер-слайд изменения отражаются на всех слайдах данной презентации.

12.9. Установка режимов демонстрации

Последним шагом в подготовке презентации является задание параметров показа презентации. Данные параметры собраны на панели *настройка демонстрации*. Здесь определяется:

- будет ли демонстрация осуществляться в автоматическом режиме или под управлением докладчика или пользователя;
- будет ли демонстрация делаться со звуковым сопровождением или без него;
- будут ли использованы назначенные эффекты анимации;
- какие слайды включить в показ;
- будет ли автоматически повторяться показ слайдов по достижению конца презентации.

Вопросы для самопроверки:

1. Перечислите варианты создания презентации. В каких случаях удобно использовать каждый из них?
2. Каким образом применить шаблон оформления: а) для одного слайда; б) для нескольких слайдов; в) для всей презентации?
3. Каким образом изменить оформление: а) одного слайда; б) нескольких слайдов; в) всей презентации?
4. Какие существуют уровни анимации слайдов?
5. Каким образом настроить анимацию слайда в целом?
6. Как добавить эффекты анимации для отдельных элементов слайда?
7. Как установить скорость смены слайдов?
8. Какие существуют режимы смены слайдов?
9. Как создать новую презентацию на основании уже имеющейся?
10. Какой размер шрифта используется для оформления: а) заголовка слайда; б) текста слайда?

Навчальне видання

Методичні вказівки
до самостійної роботи
з дисципліни

«ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ В ОХОРОНІ НАВКОЛИШНЬОГО СЕРЕДОВИЩА»

**Модуль 2. «Програмні засоби для обробки екологічної інформації.
Засоби пред'явлення і отримання інформації»**

(для студентів 2 курсу (4 семестр) денної форми навчання
напрямку підготовки 6.040106 «Екологія, охорона навколишнього
середовища і раціональне природокористування»)

(рос. мовою)

Укладачі **ПОНОМАРЕНКО** Євгеній Георгійович,
ЛОМАКІНА Ольга Сергіївна

Комп'ютерне верстання *Ю. Ю. Конюшенко*

План 2010, поз. 54М

Підп. до друку 06.05.2010 р.	Формат 60x84/16
Друк на ризографі.	Ум. друк. арк. 2,4
Зам. №	Тираж 50 пр.

Видавець і виготовлювач:
Харківська національна академія міського господарства,
вул. Революції, 12, Харків, 61002
Електронна адреса: rectorat@ksame.kharkov.ua
Свідоцтво суб'єкта видавничої справи:
ДК № 4064 від 12.05.2011 р.